

```
--
-- file: NMTrpMgr.mib
--
-- COMTEK Services, Inc.
-- MIB for VMS NM*TrpMgr Subagent
-- Release      3.5
-- Date        January 2002
-- Author      NF
--
-- Copyright 1996-2002 COMTEK Services, Inc. All Rights Reserved.
--
-- This COMTEK Services VMS NM*TrpMgr SNMP Management Information Base
-- Specification (MIB) embodies COMTEK Services' confidential and
-- proprietary intellectual property. COMTEK Services retains all
-- title and ownership in the Specification, including any
-- revisions.
--
-- This Specification is supplied "AS IS," and COMTEK Services makes
-- no warranty, either express or implied, as to the use,
-- operation, condition, or performance of the Specification.
--
```

```
COMTEK-VMS-NM-TRPMGR-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    DisplayString          FROM RFC1213-MIB
    OBJECT-TYPE            FROM RFC-1212
    TimeTicks, Counter     FROM RFC1155-SMI
    comtekVmsNMTrpMgrMib  FROM COMTEK-DEFINITIONS-MIB;
```

```
tTrap OBJECT IDENTIFIER ::= { comtekVmsNMTrpMgrMib 1 }
tSwErrInfo OBJECT IDENTIFIER ::= { comtekVmsNMTrpMgrMib 2 }
tCfg OBJECT IDENTIFIER ::= { comtekVmsNMTrpMgrMib 3 }
```

```
-- The NM*TrpMgr Subagent process maintains an internal trap table
-- containing the latest traps that have been sent by the subagent.
-- The number of traps that are retained in the internal trap table
-- is controlled by the tCfgTrapTblSize variable. Each trap that is
-- sent by NM*TrpMgr includes a trap sequence number and a timestamp
-- as well as any trap specific data. The trap sequence number may
-- be used to request that a trap be resent to the manager.
--
```

```
-- The number of traps per second that may be sent is governed by the
-- tCfgMaxTrapSec variable. tTrapLastSeqNumSent identifies the
-- sequence number of the last trap that was actually sent, if no
-- backlog of traps exists, this number will be one less than
-- tTrapNextSeqNum. If a backlog exists due to throttling, then
-- the size of the backlog may be determined by comparing these
-- numbers. If the size of the backlog exceeds the size of the
-- trap table (as set by tCfgTrapTblSize), the backlogged traps
-- will be lost as they are overwritten by new traps, the number
-- of traps that have been lost is contained in tTrapLostCount.
```

```
tTrapNextSeqNum OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
```

STATUS mandatory
DESCRIPTION
"The next sequence number in the trap table. When this variable is sent as part of a trap, it identifies the sequence number assigned to that trap. This is a 32-bit value and will wrap after 4294967295."
::= { tTrap 1 }

tTrapTime OBJECT-TYPE
SYNTAX TimeTicks
ACCESS read-only
STATUS mandatory
DESCRIPTION
"tCfgUpTime when the trap was created."
::= { tTrap 2 }

tTrapResendSeqNum OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Sequence number of a trap message to be resent to the manager. Setting this variable to a trap sequence number causes the specified trap message to be resent. If the set-request contains a sequence number that is not available (the number of items retained in the trap table is dependent on the value of tCfgTrapTblSize), an error indicator will be returned."
::= { tTrap 3 }

-- User trap messages may be used to send textual data to the SNMP manager.
-- User traps can accomodate long messages - up to 1785 characters. Due
-- to restrictions in SNMP, a single variable may only contain 255 characters
-- so user trap messages which contain more than 255 characters are broken
-- into 255 character chunks. See variables prefixed tSev below for severity
-- coded user traps.

tTrapUserMsgOne OBJECT-TYPE
SYNTAX DisplayString (SIZE (0..255))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Part 1 of user trap data. If the user message is longer than 255 characters, the remainder will be sent in tTrapUserMsgTwo, tTrapUserMsgThree, tTrapUserMsgFour, tTrapUserMsgFive, and tTrapUserMsgSix, and tTrapUserMsgSeven as appropriate."
::= { tTrap 4 }

tTrapUserMsgTwo OBJECT-TYPE
SYNTAX DisplayString (SIZE (0..255))
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Part 2 of user trap data."
::= { tTrap 5 }

```

tTrapUserMsgThree OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Part 3 of user trap data."
    ::= { tTrap 6 }

tTrapUserMsgFour OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Part 4 of user trap data."
    ::= { tTrap 7 }

tTrapUserMsgFive OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Part 5 of user trap data."
    ::= { tTrap 8 }

tTrapLastSeqNumSent OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Sequence number of the last trap sent. This value may be
        compared to tTrapNextSeqNum to determine if a backlog of traps
        exists. If this value is one less than tTrapNextSeqNum, then
        no backlog exists. A backlog of traps may exist if more traps
        are generated per second than may be sent (as specified by the
        variable tCfgMaxTrapSec)."
    ::= { tTrap 9 }

tTrapLostCount OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "This variable indicates the number of traps that have been
        lost. A trap is considered to be lost if it is overwritten
        in the internal trap table before it is sent to the NMS.
        This may occur if the trap table is not large enough (see
        tCfgTrapTblSize) to hold all of unsent traps. How quickly
        traps are sent to the NMS is governed by the tCfgMaxTrapSec
        variable. To prevent trap loss, set these two variables
        so that the trap backlog does not overflow the trap table
        and so that traps do not flood the network."
    ::= { tTrap 10 }

tTrapUserMsgSix OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-only

```

```

STATUS mandatory
DESCRIPTION
    "Part 6 of user trap data."
::= { tTrap 11 }

tTrapUserMsgSeven OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "Part 7 of user trap data."
    ::= { tTrap 12 }

-- Below are variables relating to the severity coded user traps.
-- Each severity coded trap contains two parts: facility/severity/
-- identifier string and the message text. NM*TrpMgr parses each
-- message placed in the severity coded mailboxes. If the
-- facility/severity/identifer string is not found at the beginning
-- of the message, then the this variable will be empty in the
-- severity coded trap message.

tSev OBJECT IDENTIFIER      ::= { tTrap 13 }

tSevSuccessCode OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "User trap message identifier with a severity of Success."
    ::= { tSev 1 }

tSevSuccessMsg OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "User trap message text with a severity of Success."
    ::= { tSev 2 }

tSevInfoCode OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "User trap message identifier with a severity of
        Informational."
    ::= { tSev 3 }

tSevInfoMsg OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "User trap message text with a severity of Informational."
    ::= { tSev 4 }

```

```
tSevWarningCode OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "User trap message identifier with a severity of Warning."
    ::= { tSev 5 }
```

```
tSevWarningMsg OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "User trap message text with a severity of Warning."
    ::= { tSev 6 }
```

```
tSevErrorCode OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "User trap message identifier with a severity of Error."
    ::= { tSev 7 }
```

```
tSevErrorMsg OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "User trap message text with a severity of Error."
    ::= { tSev 8 }
```

```
tSevFatalCode OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "User trap message identifier with a severity of Fatal."
    ::= { tSev 9 }
```

```
tSevFatalMsg OBJECT-TYPE
    SYNTAX DisplayString (SIZE (0..255))
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "User trap message text with a severity of Fatal."
    ::= { tSev 10 }
```

```
-- When a software error is encountered, a tSwErr trap is generated containing
-- the following software error description items. Invalid sets of the
-- tCfgFile variable will cause a tSwErr trap to be generated to explain the
-- error.
```

```
tSwErrStatus OBJECT-TYPE
    SYNTAX INTEGER
    ACCESS read-only
```

```
STATUS mandatory
DESCRIPTION
    "The latest software status code."
::= { tSwErrInfo 1 }
```

```
tSwErrFile OBJECT-TYPE
SYNTAX DisplayString (SIZE (0..255))
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The file name of the software element which generated the
    latest software status code."
::= { tSwErrInfo 2 }
```

```
tSwErrLineNum OBJECT-TYPE
SYNTAX INTEGER
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "The line number in the file which generated the latest
    software status code."
::= { tSwErrInfo 3 }
```

```
tSwErrLastTime OBJECT-TYPE
SYNTAX TimeTicks
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "tCfgUpTime when the most recent software status code
    was generated."
::= { tSwErrInfo 4 }
```

```
tSwErrMessage OBJECT-TYPE
SYNTAX DisplayString
ACCESS read-only
STATUS mandatory
DESCRIPTION
    "Description of the latest software error."
::= { tSwErrInfo 5 }
```

```
-- Configuration parameters which are read from disk and may be
-- viewed/changed by the network manager. All changes to these
-- parameters are written to the disk file.
```

```
tCfgFile OBJECT-TYPE
SYNTAX DisplayString (SIZE (0..255))
ACCESS read-write
STATUS mandatory
DESCRIPTION
    "Name of the file currently being used for NM*TrpMgr
    initialization parameters. This file is identified by the
    logical name COMTEK$TRPMGR_CONFIG. Changing this variable
    redefines the COMTEK$TRPMGR_CONFIG logical name. NM*TrpMgr
    must be reinitialized (by setting the variable
    tCfgReinitSubagent to true) before the new configuration
    file is used. If a set-request is performed on this variable
    and the specified file name does not exist, is not accessible,
    or contains invalid data, the file name is not changed, an
```

error is returned, and a tSwErr trap is generated describing the error.

Set-requests: Take effect only after reinitialization
(tCfgReinitSubagent)

NOTE: To make changes to this value permanent across system reboots, the definition of this logical name must also be changed in the system startup procedures."

::= { tCfg 1 }

tCfgLogFile OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..255))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"Name of the log file currently being used by NM*TrpMgr. If this process is run interactively, this variable will have the value SYS\$OUTPUT: (standard output)."

::= { tCfg 2 }

tCfgReinitSubagent OBJECT-TYPE

SYNTAX INTEGER { true(1), false(2) }

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Flag which may be set to force the NM*TrpMgr Subagent to reinitialize. Reinitialization involves resetting the process configuration using the parameters contained in the tCfgFile and sending the tWarmStart trap."

DEFVAL { false }

::= { tCfg 3 }

tCfgTraps OBJECT-TYPE

SYNTAX INTEGER { enabled(1), disabled(2) }

ACCESS read-write

STATUS mandatory

DESCRIPTION

"Flag which determines if any trap messages are sent to the NM*Master Agent. Setting this variable to disabled(2) disables all traps from being sent.

Configuration Variable: Traps

Set-requests: Take effect immediately

Changes to this value are written to the tCfgFile."

DEFVAL { enabled }

::= { tCfg 4 }

tCfgTrapTblSize OBJECT-TYPE

SYNTAX INTEGER (10..5000)

ACCESS read-write

STATUS mandatory

DESCRIPTION

"The maximum number of traps to be retained. Traps that are retained may be resent (see tTrapResendSeqNum) unless

too many subsequent traps have been generated. This variable controls how many traps are retained in NM*TrpMgr's internal trap table.

Configuration Variable: TrapTableSize
Set-Requests: Take effect immediately

Changes to this value are written to the tCfgFile."

DEFVAL { 100 }
::= { tCfg 5 }

tCfgHostName OBJECT-TYPE

SYNTAX DisplayString (SIZE(0..255))
ACCESS read-only
STATUS mandatory
DESCRIPTION

"The name of the host that the subagent is communicating with. This is the host name where the NM*Master Agent resides. This variable is not currently used.

Configuration Variable: Host"

DEFVAL { "localhost" }
::= { tCfg 6 }

tCfgTimeout OBJECT-TYPE

SYNTAX INTEGER (0..3600)
ACCESS read-only
STATUS mandatory
DESCRIPTION

"This variable identifies the time-out value that is to be used by the subagent during subtree registration. If this value is set to zero, the default value (saDefaultTimeout) contained in the NM*Master Agent will be used. If the value specified exceeds the current maximum that the NM*Master Agent will permit (saMaxTimeout), the maximum value will be used. This value is specified in seconds.

Configuration Variable: Timeout"

DEFVAL { 0 }
::= { tCfg 7 }

tCfgControlTermProc OBJECT-TYPE

SYNTAX INTEGER { enabled(1), disabled(2) }
ACCESS read-only
STATUS mandatory
DESCRIPTION

"Flag which may be used to disable process termination by the tCfgTermProc variable. Setting this variable to enabled(1) allows the tCfgTermProc variable to be set to true(1) to terminate the NM*TrpMgr process. Setting this variable to disabled(2) prevents the tCfgTermProc variable from being set.

Configuration Variable: ControlTermProc"

DEFVAL { disabled }
::= { tCfg 8 }

tCfgTermProc OBJECT-TYPE

SYNTAX INTEGER { true(1), false(2) }
ACCESS read-write
STATUS mandatory
DESCRIPTION
"Flag which may be used to shutdown the NM*TrpMgr process.
Setting this flag to true causes the NM*TrpMgr process to
respond to the set-request, send the tTermProc trap, close
the connection to the NM*Master Agent, and shutdown. If the
tCfgControlTermProc flag state is disabled(2), the tCfgTermProc
variable can not be set.

Set-requests: Take effect immediately if
tCfgControlTermProc is enabled(2)"

DEFVAL { false }
::= { tCfg 9 }

tCfgVersion OBJECT-TYPE

SYNTAX DisplayString
ACCESS read-only
STATUS mandatory
DESCRIPTION

"Current version of the NM*TrpMgr Subagent software."
::= { tCfg 10 }

tCfgUpTime OBJECT-TYPE

SYNTAX TimeTicks
ACCESS read-only
STATUS mandatory
DESCRIPTION

"Subagent up time."
::= { tCfg 11 }

tCfgMaxTrapSec OBJECT-TYPE

SYNTAX INTEGER (0..100)
ACCESS read-write
STATUS mandatory
DESCRIPTION

"This value specifies the maximum number of traps per second
that may be sent. A value of zero disables trap throttling,
allowing all traps to be sent as they are generated. It
is recommended that this value and tCfgTrapTblSize be
tuned to each operational environment.

Configuration Variable: MaxTrapSec
Set-requests: Take effect immediately.

Changes to this value are written to the tCfgFile."

DEFVAL { 0 }
::= { tCfg 12 }

END